REMARKS

Claims 1-13 and 16-19 have been canceled in previous amendments. Claims 14, 15 and 20-25 were rejected under 35 USC 112, first paragraph. Claims 14 and 22 were rejected under 35 USC 103 based on Holt et al in view of Smullen et al. Claims 15 and 23 were rejected under 35 USC 103 based on Holt et al. in view of Smullen et al and Applicant Admitted Prior Art (AAPA). Claim 20 was rejected under 35 USC 103 based on Shin et al. in view of Smullen et al, Holt et al. and Kubo. Claim 21 was rejected under 35 USC 103 based on Shin et al, Smullen et al., Holt et al., Kubo and AAPA. Claim 24 was rejected under 35 USC 103 based on Shin et al, in view of Smullen et al and Kubo. Claim 25 was rejected under 35 USC 103 based on Shin, Smullen, Kubo and AAPA.

Claims 14, 15 and claims 20-25 have been canceled above, and new claims 26-31 have been presented. Applicants respectfully traverse the foregoing rejections as applied to the new claims 26-31, based on the following.

35 USC 112, first paragraph Rejection

Claim 26 has been amended to recite both the slow response and nonresponsive conditions, and the respective corrective actions. The nonresponsive condition is recited as means, responsive to said one server not handling said request by said second predetermined time-out, for automatically initiating a memory dump of said one server without determining if said one server is currently operational. This is supported by the following: "Program 30 determines if the request is still pending at the end of the "X" time-out, i.e. a response has not yet been received from the target server and the "No Server Contact" return code has not been received from server 34 (decision 130). If the request is still pending, then program 30 (parent thread) sets a return code to "Server Non Responsive" to indicate a problem with the target server (step 134)." Page 7 lines 24-28. "If the latest return code is "Non Responsive", then local routine 70 will request a memory dump from the target server". Page 9 lines 18-19. The "slow response" condition and corrective action is supported on Page 8 lines 24-26 and Page 9 lines 12-15. Therefore, claim 26 complies with 35 USC 112, first paragraph. Claims 29 and 32 similarly comply with 35 USC 112, first paragraph.

35 USC 103 Rejections Based on All Cited Prior Art

Claim 26 recites a system for managing a plurality of servers in a cluster. A request is sent to one of the servers. Responsive to the one server handling the request after a first predetermined time-out but before a second, greater predetermined time-out, a count is incremented. Responsive to the one server not handling the request by the second predetermined time-out, the count is incremented without determining if the one server is currently operational. If the incremented count is less than a predetermined integer threshold greater than one, no corrective action is taken. If the incremented count equals the threshold and the one server handles the request after the first predetermined time-out but before the second predetermined time-out, a dispatcher for the one server is automatically notified to reduce a rate of dispatching new requests to the one server. If the incremented count equals the threshold and the one server does not handle the request by the second predetermined time-out, a memory dump of the one server is automatically initiated.

The Examiner cited Smullen et al. against a feature of original claim 17 which involved a memory dump. Smullen et al. disclose "to dump the memory before reloading or rebooting the processor", as part of a corrective action when the processor has halted and needs to be reloaded or rebooted:

> "the fast memory dump process is triggered by the detection of a CPU halt 100. Upon detection of the CPU halt, the RCVDUMP program will be called 102 with the PRTMEM and RELOAD options. RCVDUMP will then send a START_DUMP call to notify the system 10, 11 that a memory dump is commencing 104." Smullen et al. Column 5 lines 27-33.

New claim 26 has multiple differences over Smullen et al. Before new claim 26 initiates the memory dump, there have to be at least two events that incremented the count. **Two different types of events can increment the count,** including a slow response to a request or no response to a request by a second predetermined time-out. Thus, a "slow" response to a request increments the count as well as no response to a request by a second predetermined time-out. Smullen et al. do not consider slow responses at all, in the decision whether to initiate a memory dump. Also, Smullen et al. do not consider no response to a request, unless the processor is determined to have failed.. So, Smullen et al. do not teach an incrementing of the count without determining if the server is operational, as recited in claim 26. Moreover, in claim 26, when the count reaches the threshold, the current state will determine the type of corrective action, i.e. reduce the rate of new requests in the case of a current slow response and a memory dump in the case of a currently non responsive server. Smullen et al. do not teach that two different types of responses can increment the same count and result in a memory dump if the current state is "non responsive" and the prior state was either "slow response" or "non responsive". In claim 26, if the current state is "slow response" when the count reaches the threshold, then something different happens (reduce rate of new requests) than if the current state is "non responsive". All of these features of claim 26 would not have been obvious in view of Smullen et al. because Smullen et al. only sense and react when the processor has halted.

Holt et al. teach that "a counter of the number of time-outs for the special purpose search resource is incremented." Column 5 lines 57-59. In contrast to claim 26, the counter of Holt et al. is counting a single type of problem with a single time-out. Therefore, Holt et al. do not teach or suggest the count of claim 26 which is incremented for both a slow response and no response. Therefore, claim 26 was not obvious in view of Smullen et al. and Holt et al.

Shin et al. teach "transmitting a health check message to a processor and counting a health check count until a response message with respect to the health check message is received, comparing the health check count with a first predetermined time to judge whether the system is in an overload state or not and accordingly performing a processor processing function, comparing the health check count with a second predetermined time to judge whether the system operates or not and accordingly performing a processor processing function, and comparing the health check count with a third predetermined time to judge whether the system is abnormal or not and accordingly performing a processor processing function." Shin et al. Column 2 lines 29-41. However, in contrast to claim 26, **Shin et al. do not teach or suggest a combined count of different conditions (slow response and no response by a second time-out)**, and when a threshold is reached in the current value of the count, a corrective action is taken based on the current condition. Instead, Shin et al. consider and act on each event independent of the other events. Therefore, Shin et al. do not teach or suggest claim 26, alone or in combination with the other references.

New, independent claims 29 and 32 distinguish over Smullen et al. for the same reason that new claim 26 distinguishes thereover.

In addition, new claim 30 recites that a third condition will also increment the same count, i.e. a nonoperational state of the server. Thus, **three different conditions can all increment the same count,** i.e. nonoperational server, "slow response" server and "non responsive" server. When the count reaches the threshold, the current state will determine the type of corrective action, i.e. a reset in the case of nonoperational state, a reduction in the rate of new requests in the case of slow response and a memory dump in the case of non responsive server. This is not taught by Smullen et al. which only sense and react when the processor has halted. Shin et al. teach restarting of a server which does not respond for a predetermined time, but does not teach the foregoing combination recited in claim 30 where three different conditions can all increment the same count. Kubo et al. teach load balancing of a slow to respond server, but do not teach the foregoing combination of claim 30 where three different conditions can all increment the same count.

New dependent claim 27 further recites:

"further comprising a network dispatcher for receiving client requests from client computers and dispatching said client requests to said servers including said one server; and

wherein **the test request and the automatic initiation of the memory dump** bypass said network dispatcher."

The prior art and AAPA do not teach this arrangement either where client requests go through a dispatcher en route to the servers, and **the test requests and the automatic initiation of the memory dump** bypass the dispatcher en route to the one server. Kubo teach,

> "a terminal monitors the response time of processing at the server, an if the response time exceeds a predetermined value, then the terminal sends a path change request to change the executing computer to another computer." Kubo Column 1 lines 43-46.

Kubo does not mention an automatic initiation of a memory dump or that it bypasses a dispatcher which forwards client requests to the servers. Smullen et al. do not disclose that both test requests and the automatic initiation of the memory dump bypass the dispatcher, whereas client requests go through the dispatcher. AAPA does not teach that the automatic initiation of the memory dump bypasses the dispatcher en route to the one server. Therefore, claim 27 was not taught or obvious in view of Kubo, Smullen et al. and/or AAPA.

Dependent claims 30 and 33 similarly distinguish over Kubo, Smullen et al. and AAPA.

Based on the foregoing, Applicants request allowance of the present patent application as amended above.

Respectfully submitted,

Dated: 7-12-2006

607-429-4368 (Phone)
607-429-4119 (Fax)

Arthur J. Samodovitz
Reg. No. 31,297